AMENDMENTS

Please amend the above-referenced application as follows:

IN THE CLAIMS:

Please amend the claims as follows.

C1

1. (Twice Amended.) A method for finding a predefined plurality of instructions, if available, that are ready to be executed and that reside in an instruction reordering mechanism of a processor that can launch execution of instructions out of order via a predefined number of ports, comprising the steps of:

(a) providing said instruction reordering mechanism having a plurality of said instructions, each said instruction port having a respective self-timed vector logic element for causing and preventing launching, when appropriate, of said instruction; and

(b) propagating a set of signals successively during a launch cycle through said self-timed vector logic elements of said instruction reordering mechanism ~~that causes said logic elements to track which of the predefined plurality of said instructions are launched and causes the selection of no more than said predefined number of ports during said launch cycle~~.

2. (Once Amended.) The method of claim 1, further comprising the step of advising each instruction port of said instruction reordering mechanism during each launch cycle either that said instruction will be launched or that said instruction will not be launched.

3. (Once Amended.) The method of claim 1, wherein said signals are propagated ~~monotonically~~ through said self-timed vector logic elements in response to only one direction of logic transition.

4. (Originally Submitted.) The method of claim 1, further comprising the step of communicating said predefined plurality of said instructions to a corresponding predefined plurality of ports associated with one or more execution resources.

5. (Once Amended.) The method of claim 1, further comprising the step of, after said predefined plurality of said instructions have been selected, propagating a lost signal to remaining self-timed vector logic elements associated with remaining instructions of said instruction reordering mechanism to indicate to said remaining self-timed vector logic elements that their respective remaining instructions have not been selected.

6. (Once Amended.) The method of claim 1, further comprising the steps of:

(c) after said predefined plurality of said instructions have been selected, propagating a lost signal to remaining self-timed vector logic elements associated with remaining instructions of said instruction reordering mechanism to indicate to said remaining self-timed vector logic elements that their respective remaining instructions have not been selected;

(d) performing steps (b) and (c) during a single cycle associated with one or more execution resources; and

(e) communicating said predefined plurality of said instructions from said instruction reordering mechanism to a corresponding predefined plurality of ports associated with said one or more execution resources.

7. (Once Amended.) The method of claim 1, further comprising the step of:

(c) providing said instruction reordering mechanism in a form of a queue having a plurality of slots, each said slot having a respective one of said self-timed vector logic elements and means for temporarily storing a respective instruction; and

(d) propagating said set of said signals successively through said slots of said queue during an execution cycle.

8. (Originally Submitted.) The method of claim 1, wherein said set comprises two or more signals.

9. (Once Amended.) The method of claim 1, further comprising the step of:

(c) causing said propagation through only a predefined number of said self-timed vector logic elements during a launch cycle.

10. (Once Amended.) A method for quickly finding a predefined plurality of instructions, if available, that are ready to be executed and that reside in a queue of a processor that can launch the execution of instructions out of order, so that the found instructions can be communicated to a corresponding predefined plurality of ports associated with one or more execution resources, comprising the steps of:

(a) providing said queue having a plurality of slots, each said slot for temporarily storing a respective instruction and launching, when appropriate, execution of said respective instruction; and

(b) propagating a set of signals successively through slots of said queue during a launch cycle that, when passed through a particular slot:

(1) selects said particular slot for launching when said particular slot is ready by asserting in said slot one or more found signals that identify one or more specific ports associated with said one or more execution resources;

(2) refrains from selecting said particular slot when said particular slot is not ready by asserting in said slot a lost signal;

(3) keeps track of how many slots have been selected during said launch cycle; and

(4) causes selection of no more than said predefined plurality of said instructions during said launch cycle; and

wherein propagating occurs in response to logic transitions in only one direction.

11. (Originally Submitted.) The method of claim 10, further comprising the step of communicating said predefined plurality of said instructions from said queue to said corresponding predefined plurality of ports associated with said one or more execution resources.

12. (Originally Submitted.) The method of claim 10, further comprising the step of (c) during said launch cycle but after said predefined plurality of said instructions have been selected, propagating a lost signal to remaining slots associated with remaining instructions of said queue to indicate to said remaining slots that their respective remaining instructions have not been selected.

13. (Twice Amended.) A system for finding a predefined plurality of instructions, if available, that are ready to be executed in a processor that can launch execution of instructions out of order, comprising:

(a) an instruction reordering mechanism for temporarily storing a plurality of said instructions; and

(b) a plurality of self-timed vector logic elements associated with said instruction reordering mechanism and associated respectively with each of said instructions in said instruction reordering mechanism for causing and preventing launching, when appropriate, of respective instructions, said self-timed vector logic elements configured to propagate a plurality of signals ~~monotonically~~ through said self-timed vector logic elements such that ~~causes~~ said self-timed vector logic elements ~~to~~ select said predefined plurality of said instructions for launching and ~~to~~ de-select any remaining instructions in response to only one direction of logic transition.

14. (Once Amended.) The system of claim 13, wherein each of said self-timed vector logic elements is configured to receive said set of signals from a previous logic element, to evaluate said set of signals to determine whether or not to launch a respective instruction, to modify states associated with said set of signals based upon whether or not said respective instruction was launched, and to propagate said set of said signals to a later logic element.

15. (Once Amended.) The system of claim 13, wherein each one of said self-timed vector logic elements is implemented in combinational logic hardware.

16. (Once Amended.) The system of claim 13, wherein each said self-timed vector logic element is configured to, after said predefined plurality of said instructions have been selected, propagate a lost signal to remaining self-timed vector logic elements associated with said remaining instructions of said instruction reordering mechanism to indicate to said remaining self-timed vector logic elements that their respective remaining instructions have not been selected.

17. (Originally Submitted.) The system of claim 13, further comprising one or more execution resources having one or more ports to receive data from said predefined plurality of said instructions.

18. (Originally Submitted.) The system of claim 17, wherein at least one of said execution resources is an arithmetic logic unit (ALU).

19. (Originally Submitted.) The system of claim 17, wherein at least one of said execution resources is a multiple accumulate unit (MAC).

20. (Originally Submitted.) The system of claim 17, wherein at least one of said execution resources is a cache.

21. (Originally Submitted.) The system of claim 13, wherein said instruction reordering mechanism is a queue.

22. (Once Amended.) The system of claim 13, further comprising:

an arbitration mechanism configured to assert a start signal to one of said self-timed vector logic elements to initiate said propagation of said set of signals.

23. (Twice Amended.) A system for finding a predefined plurality of instructions, if available, that are ready to be executed and that reside in a queue of a processor that can launch execution of instructions out of order, comprising:

(a) queue means for storing a plurality of said instructions, said queue means having a plurality of launch logic means for causing and preventing launching, when appropriate, of a respective instruction; and

(b) logic means associated with said queue, said logic means for propagating ~~during a launch cycle~~ a set of signals ~~monotonically~~ to successive launch logic means to indicate both when and which of one or more ports of one or more execution resources are available for each said instruction and when none of said ports are available, wherein said means for propagating is responsive to logic transitions in only one direction.